IN THE CLAIMS

Please amend Claims 19, 25, 31-35, 40, 44, 45 and 48, and add claims 50-59 as follows:

1. – 18. (Cancelled)

19. (Currently amended) In a processing apparatus adapted to fetch plural instructions in parallel and ~~dispatches~~ to dispatch one or more subsets thereof for subsequent pipelined processing by one or more respective execution units, a method comprising:

reading a branch instruction into an instruction processing pipeline;

determining a branch target address associated with said branch instruction;

evaluating whether the branch target address corresponds to a ~~target pipeline~~ condition ~~whereby~~ where the branch target instruction is already in the instruction processing pipeline;

if the evaluation results in the negative, causing an instruction located at the branch target address to be fetched into a stage of the instruction pipeline from a cache memory using a regular fetch-from-cache sequence to service the branch; and

if the evaluation results in the affirmative, causing a shorted branch sequence to be executed to cause execution to branch to the branch target instruction.

20. (Previously presented) The method of Claim 19, wherein the branch target instruction is fetched into a dispatch stage of the instruction processing pipeline.

21. (Previously presented) The method of Claim 19, wherein the branch target instruction is fetched into a decode stage of the instruction processing pipeline.

22. (Previously presented) The method of Claim 19, wherein the branch target instruction is stored in the cache memory as a least partially pre-decoded instruction.

23. (Previously presented) The method of Claim 19, wherein the cache memory is a branch cache.

24. (Previously presented) In a processor comprising an instruction pipeline having a plurality of pipeline stages and a cache memory that holds branch target instructions, a method of controlling said cache memory, comprising:

providing an indication when a branch instruction is detected in a selected one of said pipeline stages;

evaluating a branch target address of said branch instruction to thereby identify a branch target instruction;

determining whether the branch target instruction has already been fetched into said pipeline;

determining a variable number of instructions that are already in the pipeline and are substantially between the branch instruction and the branch target instruction; and

causing the variable number of instructions already in the pipeline to be skipped over so that execution can continue at the branch target instruction.

25. (Currently amended) The method of Claim 24, wherein the branch target instruction is stored in the cache memory as a pre-decoded instruction, the evaluating comprises accessing a pre-computed branch target address value, and the branch target instruction is fetched into a stage that follows a ~~decode~~ first stage of the instruction pipeline.

26. (Previously presented) The method of Claim 24, wherein the cache memory comprises a branch cache.

27. (Previously presented) A processing apparatus comprising:

an instruction pipeline comprising a plurality of stages, said stages comprising at least an instruction dispatch stage, a decode stage, and an execution stage;

a branch control unit that analyzes instructions in the instruction pipeline and determines whether at least one branch instruction is present, and if so, determines a branch target address therefor and determines whether the branch target address corresponds to the address of an instruction that is currently in a subsequent pipeline stage, and if so, causes a set of one or more instructions already in the pipeline to be skipped over so that execution can continue at the branch target instruction that is already in the instruction pipeline.

28. (Previously presented) The processing apparatus of Claim 27, further comprising a cache memory that stores instructions and, when the branch control unit determines a branch target instruction needs to be fetched into the pipeline, causes the cache memory to couple the branch target instruction into a stage of the instruction pipeline.

29. (Previously presented) The processing apparatus of Claim 28, wherein the branch target instruction is stored in the cache memory as a pre-decoded instruction and is fetched into a stage that follows the decode stage of the instruction pipeline.

30. (Previously presented) The processing apparatus of Claim 28, wherein the cache memory is a branch cache.

31. (Currently amended) The processing apparatus of Claim 28, wherein the cache memory stores the branch target address as a precomputed branch target address and the branch control unit receives an early indication from said instruction pipeline prior to the execution of the branch instruction, wherein the branch control unit uses the precomputed branch target address to look up the branch target instruction before the branch instruction reaches the execution stage of the instruction pipeline.

32. (Currently amended) The processing apparatus of Claim 27, wherein the branch control unit receives an early indication from said instruction pipeline prior to the ~~execution~~ decoding of said branch instruction.

33. (Currently amended) In a processing apparatus which fetches plural instructions in parallel and dispatches one or more subsets thereof for subsequent pipelined processing by one or more respective execution units, a method comprising:

reading a branch instruction into an instruction processing pipeline ~~stage~~;

determining a branch target address associated with said branch instruction;

determining that a branch target instruction is located in a cache memory, wherein the branch target instruction is an instruction whose address corresponds to the branch target address, and wherein the branch target instruction is stored in pre-decoded form in the cache memory;

fetching the branch target instruction from the cache memory in pre-decoded form into a pipeline stage that follows a ~~decode~~ first pipeline stage in the instruction pipeline, to thereby reduce the number of pipeline stall cycles that would occur if the branch target instruction were to be fetched into [[a]] the first pipeline stage into which non-branch target instructions are ~~normally~~ sometimes fetched.

34. (Currently amended) The ~~processing apparatus~~ method of Claim 33, wherein the cache memory comprises a branch cache.

35. (Currently Amended) In a processing apparatus which fetches instructions into one or more instruction pipelines, a method comprising:

reading a branch instruction into an instruction processing pipeline stage;

evaluating a set of bits to determine whether the instruction is a branch instruction, and if so, to also determine to which of a plurality of types the branch instruction conforms;

based on the evaluation, generating an early branch indication signal;

in response to the early branch indication signal, accessing a pre-computed branch target address, and using the pre-computed branch target address to cause at least one branch target instruction to be fetched from a cache memory before the a branch condition associated with the branch instruction has been resolved;

performing a branch prediction and subsequently determining the prediction was incorrect; and

in response to the detection of the incorrectness of the branch prediction, causing the at least one branch target instruction to be selectively discarded to avoid execution. ~~using the pre-computed branch target address to cause at least one branch target instruction to be fetched from a cache memory.~~

36. (Previously presented) The method of Claim 35, wherein the branch target instruction is fetched into a dispatch stage of the instruction pipeline.

37. (Previously presented) The method of Claim 35, wherein the branch target instruction is fetched into a decode stage of the instruction pipeline.

38. (Previously presented) The method of Claim 35, wherein the branch target instruction is stored in the cache memory as a pre-decoded instruction and is fetched into a stage that follows a decode stage of the instruction pipeline.

39. (Previously presented) The method of Claim 35, wherein the cache memory comprises a branch cache.

40. (Currently amended) In a processing apparatus which fetches instructions into one or more instruction pipelines, a method comprising:

reading an instruction into a stage of an instruction processing pipeline;

evaluating a set of bits to determine whether the instruction is a branch instruction, and if so, to also determine to which of a plurality of types the branch instruction conforms;

based on the evaluation, generating an early branch indication signal before the branch instruction reaches ~~an execute~~ a decode stage of the instruction pipeline; and

in response to the early branch indication signal, and in response to an indication of the type of branch instruction detected, determining whether a short branch into the pipeline or a long branch out of the pipeline will be needed in the event of a mis-predicted branch.

41. (Previously presented) In a processing apparatus which fetches instructions into one or more instruction pipelines, a method comprising:

reading an instruction into a first stage of an instruction processing pipeline;

evaluating a set of bits to determine whether the instruction is a branch instruction, and if so, to also predict a branch direction; and

in the event of a mis-prediction, reading a branch target instruction out of a cache memory into a pipeline stage that is subsequent to the first pipeline stage to reduce the number of pipeline stalls that would otherwise be associated with the mis-prediction.

42. (Previously presented) The method of Claim 41, wherein the branch target instruction is stored in the cache memory as a pre-decoded instruction and is fetched into a stage that follows a decode stage of the instruction pipeline.

43. (Previously presented) The method of Claim 41, wherein the cache memory comprises a branch cache.

44. (Currently amended) In a processing apparatus adapted to fetch instructions into one or more instruction pipelines, a method comprising:

reading an instruction into a stage of an instruction processing pipeline;

evaluating a set of bits to determine whether the instruction is a branch instruction, and if so, to also predict a branch direction; and

in the event ~~of a mis-prediction~~ that a branch target instruction has already been fetched into the pipeline, branching to [[a]] the branch target instruction that is already in the pipeline ahead of the branch instruction to reduce number of pipeline stalls that would otherwise be associated with the mis-prediction.

45. (Currently amended) In a processing apparatus adapted to fetch instructions into one or more instruction pipelines, a method comprising:

reading a set of instructions and a set of previous instructions into an instruction processing pipeline ~~stage~~;

evaluating a set of bits to determine whether the set of instructions includes a branch instruction, and if so, to also predict a branch direction;

based on the evaluation, determining that a branch target instruction associated with said branch instruction has already been fetched into the instruction pipeline and in response thereto, skipping a subset of fetched instructions, said fetched instructions being disposed between the branch instruction and the branch target instruction, so that the subset of fetched instructions do not consume pipeline resources, thereby reducing a delay associated with the branch relative to branch instructions whose branch target instruction is not presently in a subsequent pipeline stage.

46. (Previously presented) In a single-chip processing apparatus adapted to fetch instructions into one or more instruction pipelines, a method comprising:

reading a set of instructions into an instruction processing pipeline;

evaluating a set of bits to determine whether the set of instructions comprises a branch instruction, and if so, to also compute a branch target address for the branch instruction;

storing the computed branch target address into a cache memory located in the single-chip processing apparatus, wherein the cache memory is configured to look up the computed branch target address in response to information related to the branch instruction;

in response to later detecting the branch instruction in the pipeline a second time, fetching the precomputed branch target address out of the cache during subsequent processing; and

causing the instruction stream to divert fetching to the branch target address using the precomputed branch target address.

47. (Previously presented) The method of Claim 46, wherein the information related to the branch instruction comprises the address of the branch instruction.

48. (Currently amended) The method of Claim 46, wherein the cache memory further stores the branch target instruction in the cache memory as a pre-decoded instruction and couples ~~into~~ the pre-decoded branch target instruction into a stage that follows a ~~decode~~ first stage of the instruction pipeline ~~in response to a branch mis-prediction~~.

49. (Previously presented) The method of Claim 46, wherein the cache memory comprises a branch cache.

50. (New) In a processor comprising an instruction pipeline having a plurality of pipeline stages and a cache memory that holds instructions including branch target instructions, a method of controlling said cache memory, comprising:

providing an early indication when a branch instruction is detected in a selected first pipeline stage that is situated before a decode stage in the instruction pipeline, wherein the branch instruction is decoded in the decode stage, and the decode stage is situated in the instruction pipeline after an instruction dispatch stage and before at least one execution stage;

looking up in a cache memory a pre-computed branch target address of said branch instruction to thereby identify a branch target instruction;

fetching both the branch target instruction and a fall-through instruction into the instruction pipeline prior to resolving whether the branch target instruction or the fall-through instruction is to be executed after the branch instruction is to be executed; and

in the event that the fall-through instruction is to be executed after the branch instruction, causing one or more branch target instructions to be skipped over so that execution can continue at the fall-through instruction.

51. (New) The method of Claim 50, wherein the branch target instruction is stored in the cache memory as at least partially pre-decoded instruction.

52. (New) The method of Claim 50, wherein the cache memory comprises a branch cache.

53. (New) A processing apparatus comprising:

an instruction pipeline comprising a plurality of stages, said stages comprising at least an instruction dispatch stage, a decode stage, and an execution stage;

a branch control unit that analyzes instructions in the instruction pipeline and determines whether a branch instruction is present, and if so, looks up a pre-computed branch target address therefor from a cache memory before the branch instruction has executed and before a branch condition associated with the branch instruction has been resolved; and

a circuit that determines whether the next instruction to be executed after the branch instruction is a fall-through instruction, and if so, causes a set of one or more branch target instructions that have already been prefetched into the pipeline to be discarded from the instruction pipeline so that execution can continue at the fall-through instruction.

54. (New) The processing apparatus of Claim 53, further comprising a cache memory that stores instructions and, wherein the branch control unit further causes the cache memory to couple the branch target instruction into a stage of the instruction pipeline.

55. (New) The processing apparatus of Claim 54, wherein the branch target instruction is stored in the cache memory as an at least partially pre-decoded instruction.

56. (New) The processing apparatus of Claim 55, wherein the cache memory is a branch cache.

57. (New) The processing apparatus of Claim 53, wherein the branch control unit causes both the branch target instructions and the fall-through instructions to be coupled into to the instruction pipeline irrespective of the resolution of the branch condition.

58. (New) In a super scalar processing apparatus which fetches instructions into an instruction pipeline, a method comprising:

reading an instruction into the instruction processing pipeline;

prior to decoding the instruction, evaluating a set of bits related to the instruction to determine whether the instruction is a branch instruction, and if so, causing both a pre-computed branch target instruction and a fall-through instruction to be coupled from a cache memory into the instruction pipeline; and

selectively executing only one of the branch target instruction and a fall-through instruction depending on a resolution of a conditional logic associated with the branch instruction.

59. (New) The processing apparatus of Claim 58, wherein the cache memory is a branch cache.